

## **Managing Software Requirements in the Context of the Scientific Enterprise**

Developing and testing production-quality software to specifications which are coupled to a scientific research program presents unique challenges. These challenges include differences between science and engineering cultures, language and priorities; the inherent unpredictability of the scientific process; and the highly complex nature of the requirements themselves.

This paper discusses the requirements management approach of a NASA Earth Observing System (EOS) instrument, the Tropospheric Emission Spectrometer (TES) project. After launch on December 2002, the TES instrument will provide the world's first three-dimensional global data set of tropospheric ozone and its precursors. The instrument will produce over 6TB of raw data per year.

Data processing will be performed by the EOS Data and Information System (EOSDIS) using software developed by the TES project. The production environment is a highly automated batch-oriented system that places a number of significant constraints on the implementation and management of the processing software. Many of the algorithms required to calibrate and analyze the data have yet to be completely developed. To accommodate these uncertainties, an incremental development approach has been adopted. Further complicating matters is the lack of previous instruments to provide precursor data of similar nature and resolution.

The TES team has developed techniques for communicating and documenting detailed specifications that involve a mixture of traditional requirements management techniques, emerging techniques previously unused in the development of science processing systems, and novel techniques unique to the project. Tool support is provided by Rational Software Inc.'s Requisite Pro requirements management software, Rational's Rose C++ CASE tool, and Platinum Technology's CCC/Harvest configuration management software, augmented by TES-developed integration and enhancements scripts.

A key factor in the TES team's success will be the integration of requirements engineering with the scientific enterprise. The divide between the domain engineering and applications engineering aspects of software development is particularly wide in the case of scientific applications.

Science algorithm developers typically have advanced degrees and many years' professional experience in a relatively narrow area of concentration. The specialized language, tools and concepts of the science team present formidable challenges to the understanding of software engineers assigned to the task. Many scientists have extensive experience in programming, and some have experience with modern languages and techniques, such as C++ and object-oriented analysis and design. However, data system development is not their primary concern, and engineering teams need to take extra steps to ensure effective communication.

Our paper will discuss in greater depth some of the issues, challenges and solutions relating to requirements management in the context of the scientific enterprise.